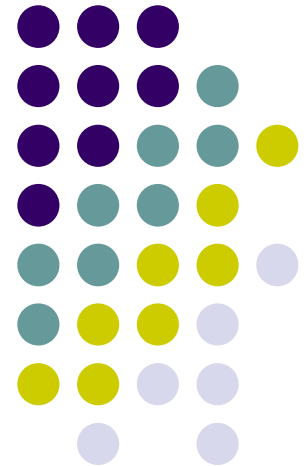


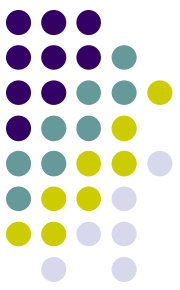
# Ingeniería de Software

---

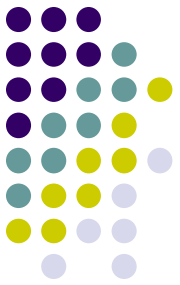
Requerimientos



# Evite Confusión en los Requerimientos



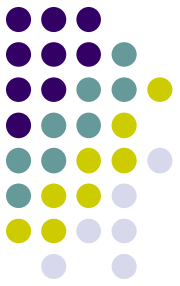
- La palabra “requerimientos” significa diferentes cosas para diferentes personas
  - Para un ejecutivo puede ser un concepto de producto de alto nivel desde el punto de vista del negocio.
  - Para un cliente puede ser una lista de ideas o soluciones propuestas.
  - Para un desarrollador puede ser una interfaz gráfica.
  - Para un patrocinador puede ser la cantidad de ganancias obtenidas posterior a la implementación.
  - Para un usuario las herramientas que le ayudarán a realizar sus tareas.



# Ejemplo

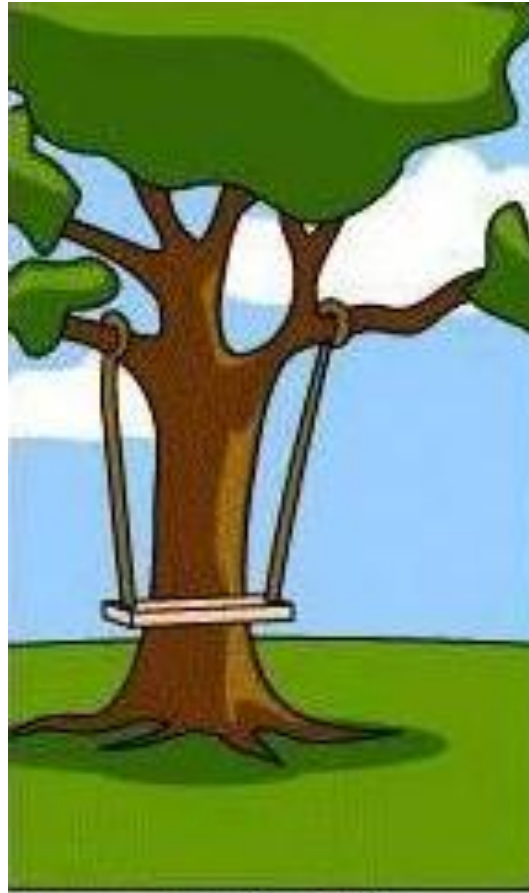
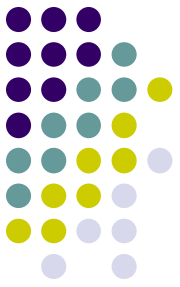
- Imagina que un usuario pide un columpio.
- Al hacer la captura de requerimientos tenemos los siguientes resultados según los diferentes puntos de vista:

# El usuario nunca sabe lo que quiere



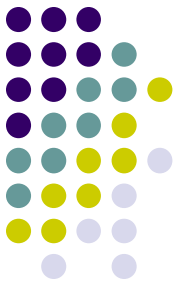
La solicitud del  
usuario

# Los líderes de proyecto son pragmáticos



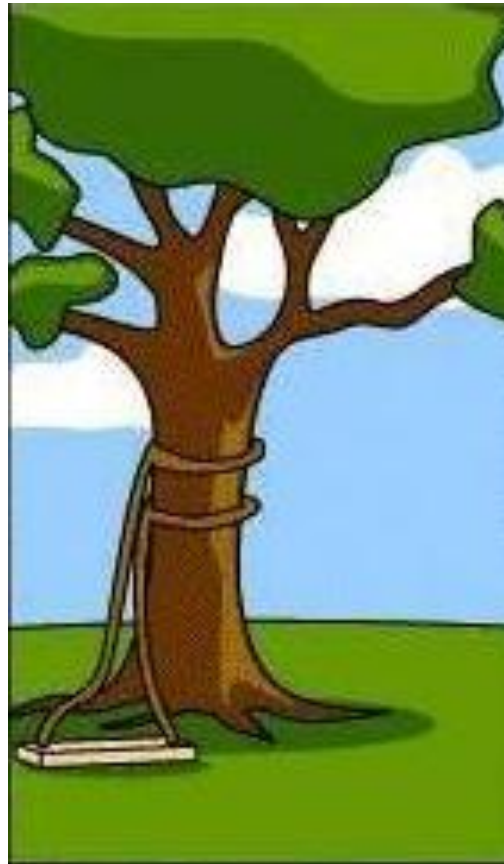
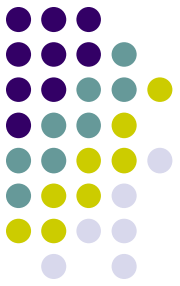
Lo que entendió el  
líder del proyecto

# El analista es sofisticado, aunque no siempre sea lo adecuado



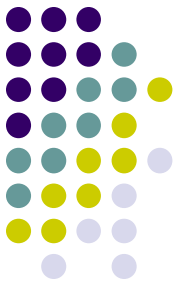
El diseño  
del analista de sistemas

# Técnicamente... cumple!



El enfoque del  
programador

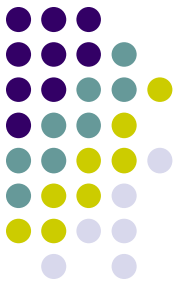
# La mercadotecnia es su eje motor.



La recomendación  
del consultor externo

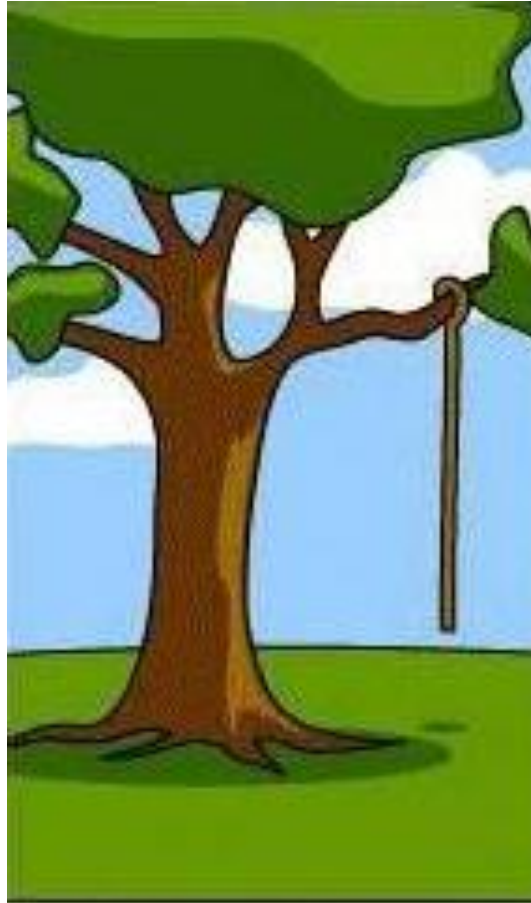
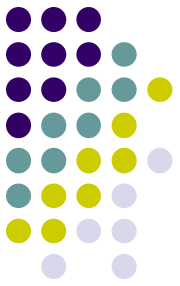


# Sin comentarios :o



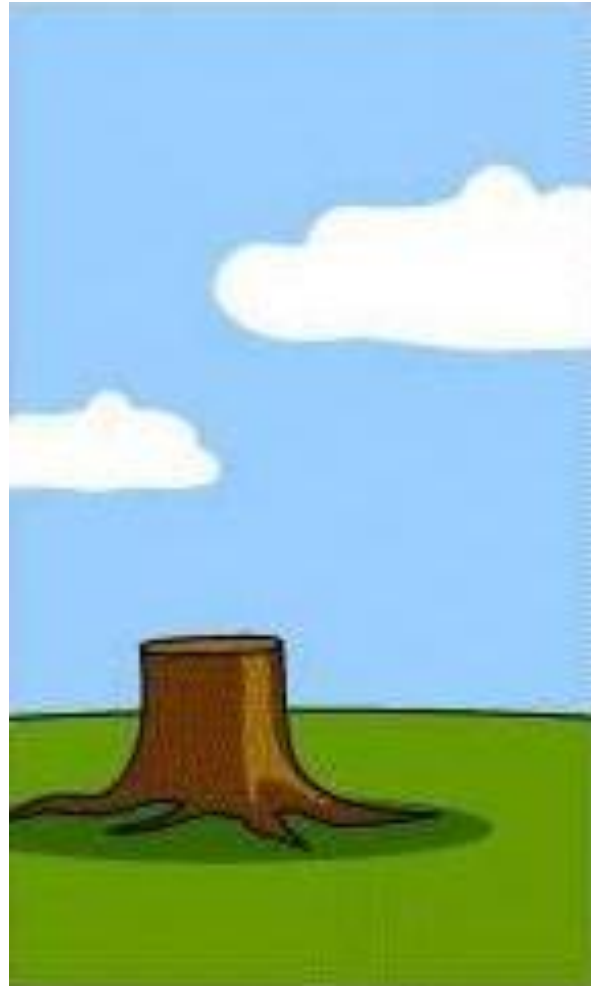
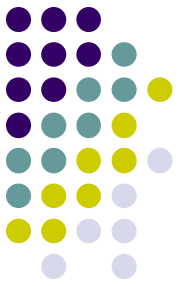
La documentación  
del proyecto

# Poca o nula calidad.



La implantación  
en producción.

# El soporte operativo





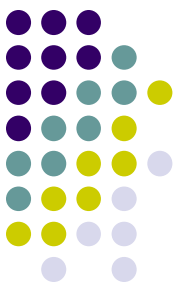
# La experiencia nos da la pauta

- Lo importante de la ingeniería de requerimientos es comprender que es lo que realmente el usuario necesita para llevar a cabo sus **tareas primordiales**. Aun a pesar del mismo usuario.



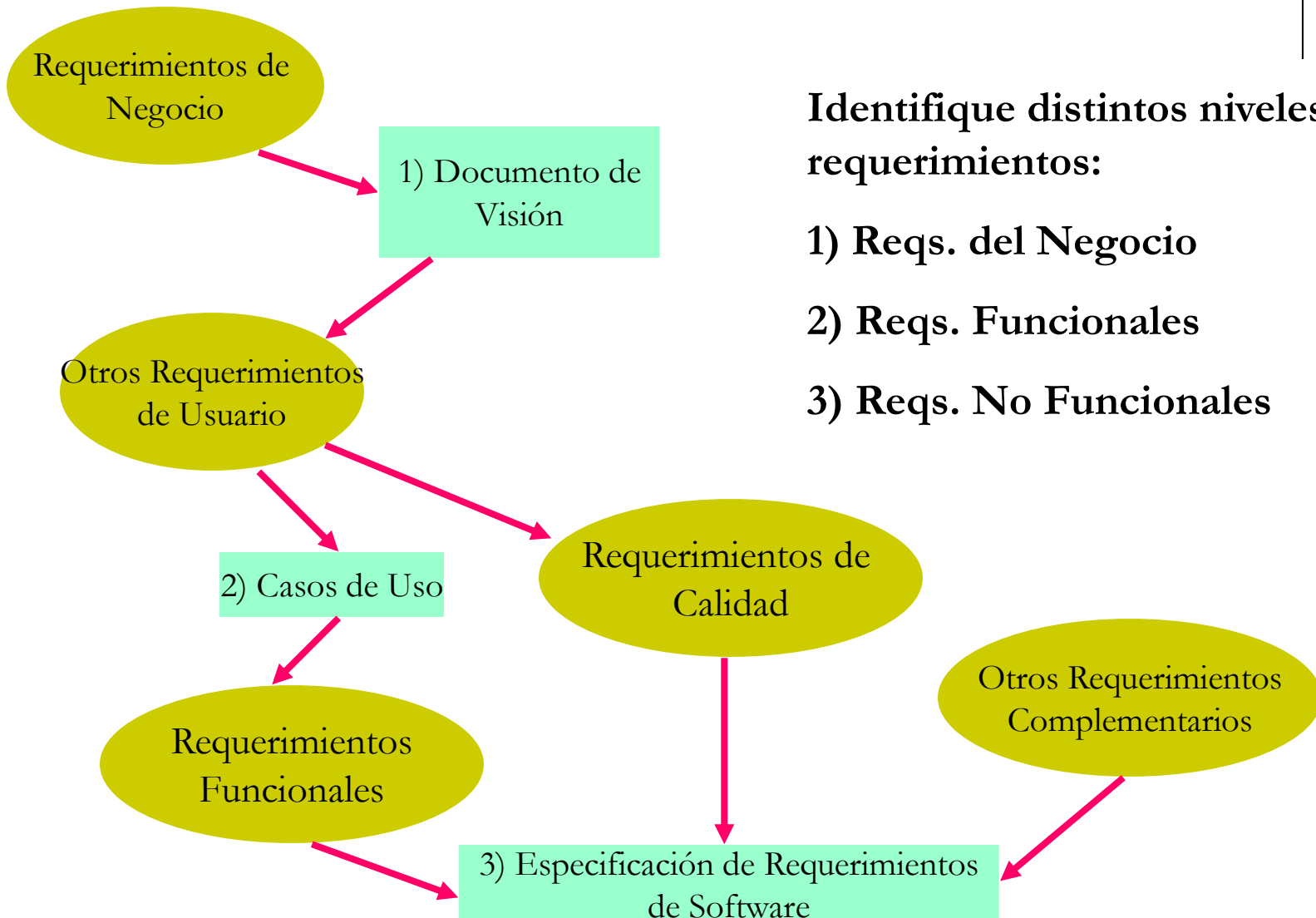
Lo que el usuario  
realmente necesitaba

# Evite Confusión en los Requerimientos cont...

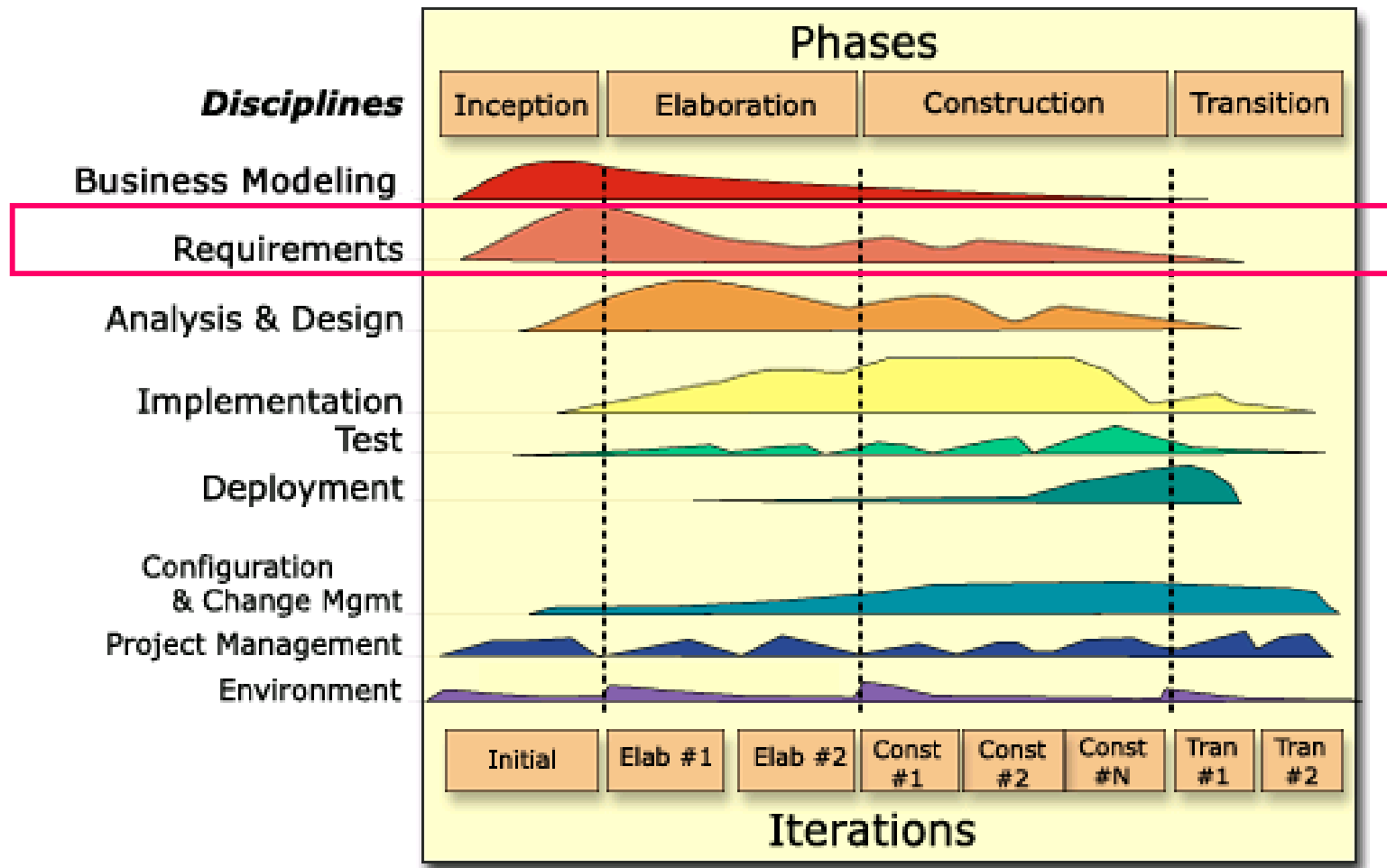


Identifique distintos niveles de requerimientos:

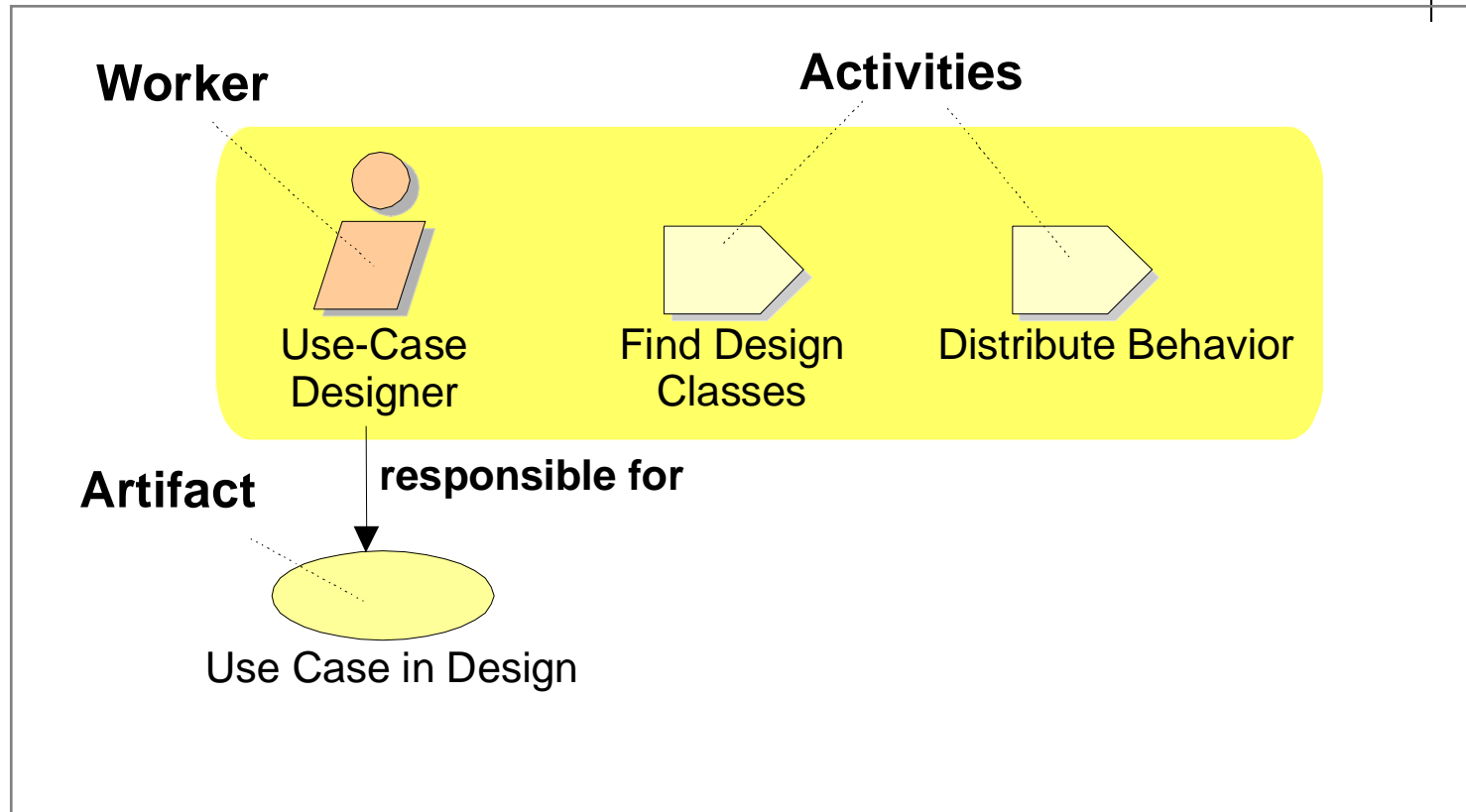
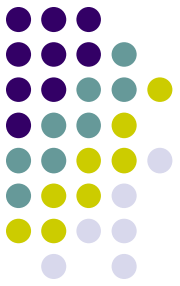
- 1) Reqs. del Negocio
- 2) Reqs. Funcionales
- 3) Reqs. No Funcionales



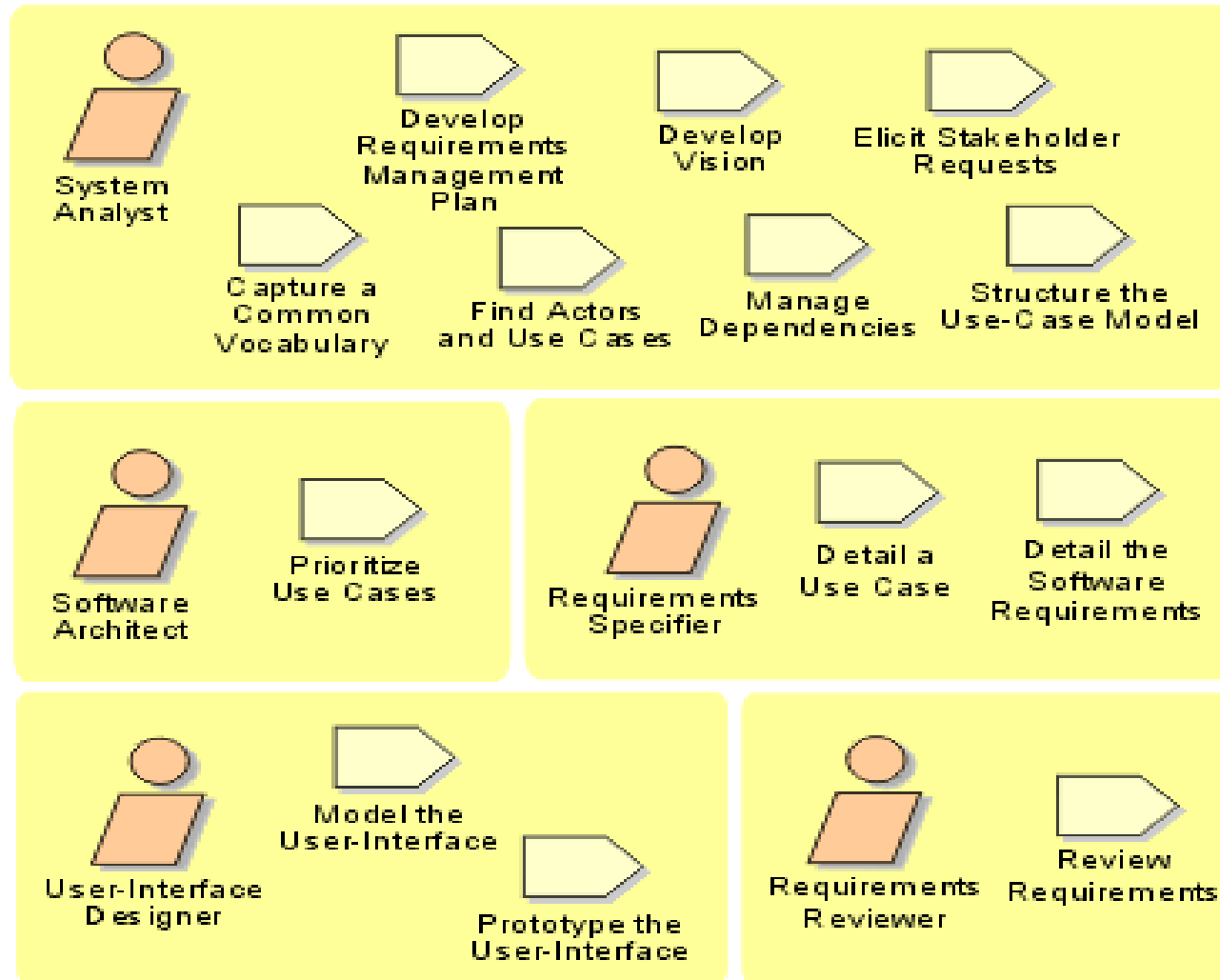
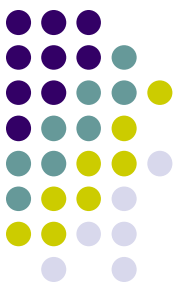
# Presentando el Proceso Iterativo: Rational Unified Process



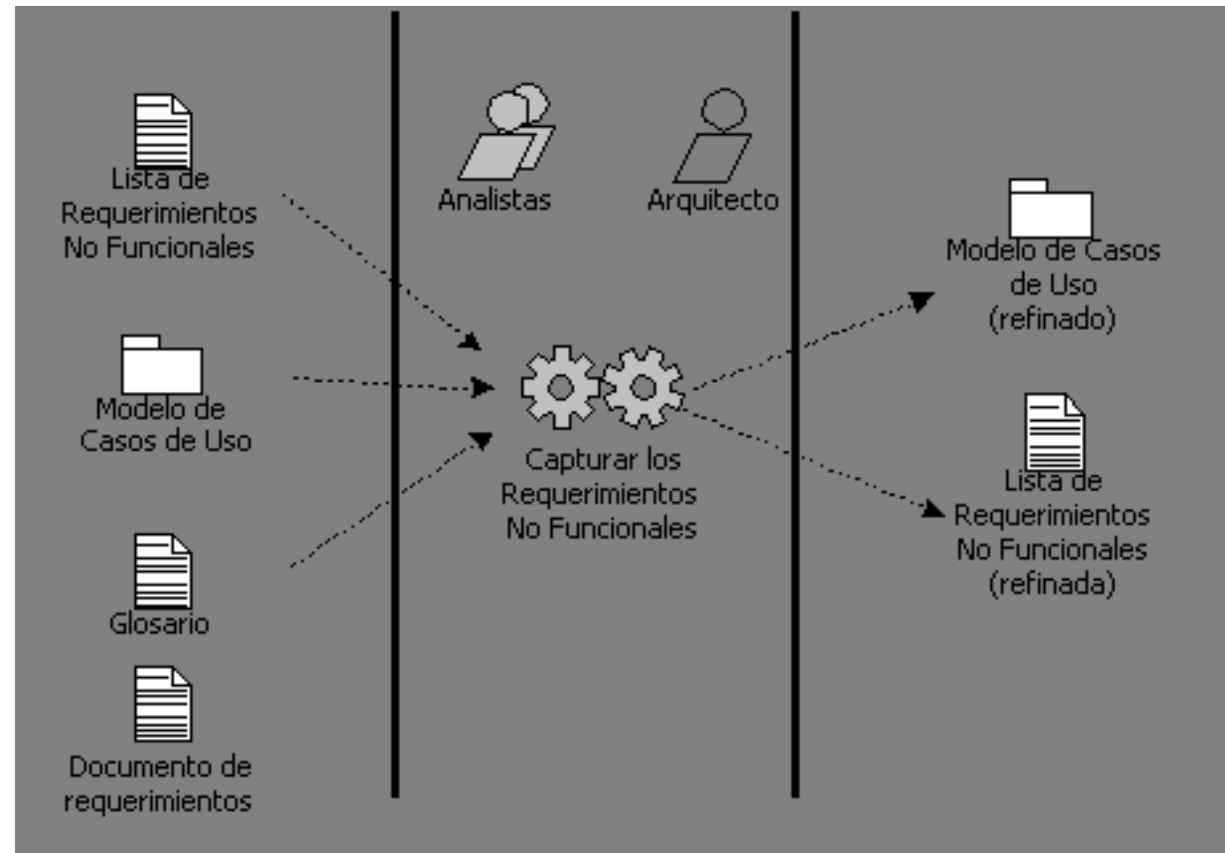
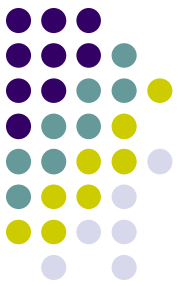
# Notación: Workers, Activities, Artifacts



# Flujos de Trabajo y Actividades por Disciplina

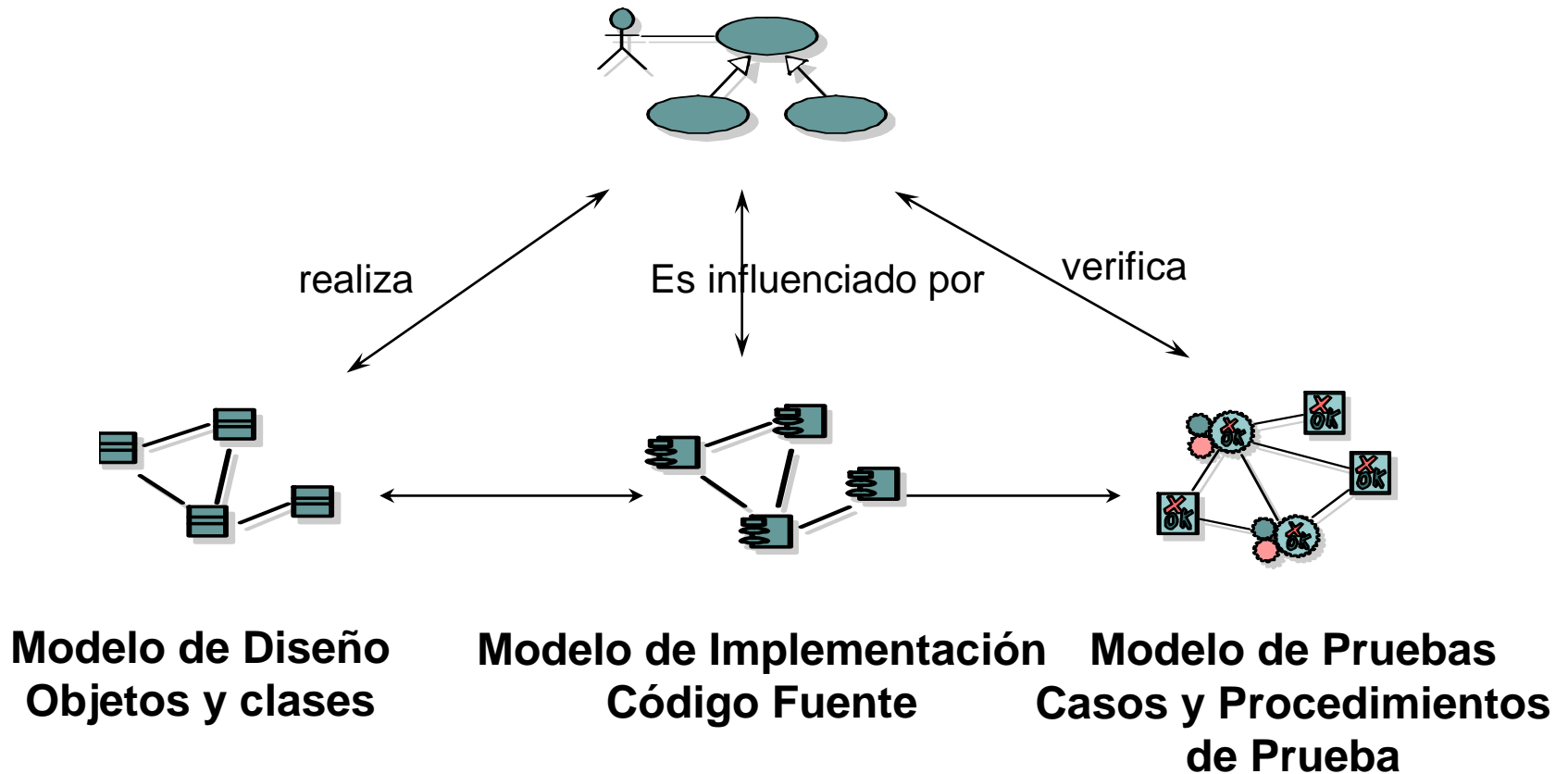
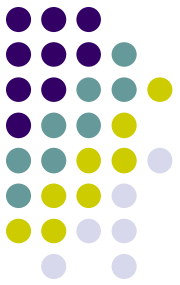




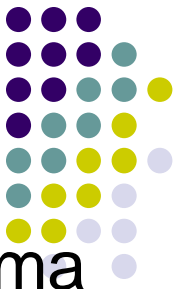


# El Enfoque de Casos de Uso

*Use-Case Model*  
*Definición del Sistema*

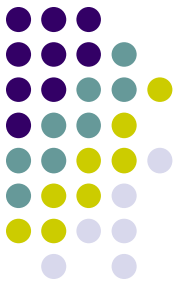


# ¿Qué es un Modelo?



- Un modelo es una descripción completa del sistema desde una perspectiva particular
  - “Completa” significa que no se necesita información adicional para entender el sistema desde esa perspectiva
- Cuatro modelos serán considerados para desarrollar nuestro sistema
  - Modelo de **Casos de Uso**
    - Describe los requerimientos desde el punto de vista del usuario
  - Modelo de **Diseño**
    - Describe las clases y objetos
  - Modelo de **Implementación**
    - Código Fuente
  - Modelo de **Prueba**
    - Casos de Prueba y Procedimientos

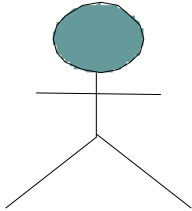
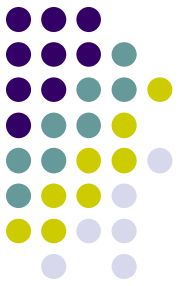
# Modelo de Casos de Uso



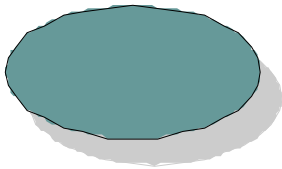
- Un *use-case model* es un modelo de las funciones de un sistema (casos de uso) y sus alrededores (actores).
- Los *actores* representan a los usuarios o cualquier sistema o dispositivo que interactúa con el sistema que se va a construir.
- Los *use cases* sirven como una secuencia unificada a través del desarrollo del sistema
- El mismo *use-case model* se usa en la captura de requerimientos, análisis, diseño y pruebas.

# Modelo de Casos de Uso

## Conceptos



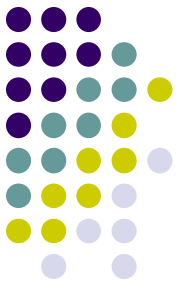
Actor



Caso de  
Uso

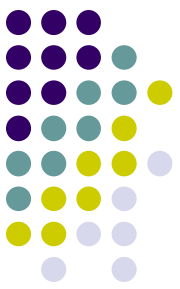
- Un *actor* representa una entidad externa que interactúa con el sistema
- Un *caso de uso* define una secuencia de acciones desarrolladas por un sistema que conducen a un resultado observable de valor a un actor

# Beneficios del Modelo de Casos de Uso

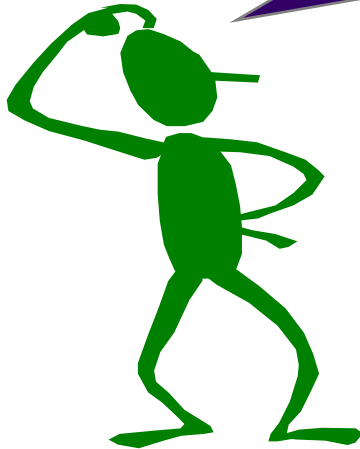


- Facilita el acuerdo con el cliente de los requerimientos del sistema
- Utiliza terminología que el usuario entiende
- Verifica el entendimiento del desarrollador del sistema
- Identifica el rol de los usuarios del sistema
- Identifica las interfases del sistema
- Ayuda a verificar que todos los requerimientos sean capturados

# El modelo ayuda a manejar la basura en los requerimientos



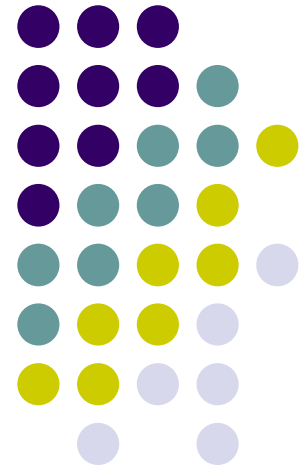
Necesitaré un nuevo caso de uso y algunos cambios a casos de uso que ya existen...



**Cliente**

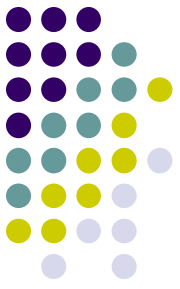
- El cliente está consciente de que hay un cambio en el modelo de casos de uso.
- Hace el impacto de costos visible para el cliente

# Casos de uso detallados





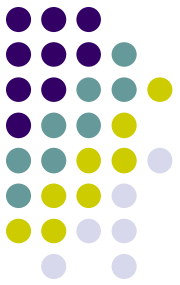
# Los casos de usos se clasifican en [Larman,1999:58]:



- **Primarios:** Representan los procesos comunes más importantes.
- **Secundarios:** Representan procesos menores o raros.
- **Opcionales:** representan procesos que pueden no abordarse.
- Según el grado de la aceptación del diseño se clasifican en **esenciales** y **reales**.

# Formato expandido de casos de uso

[Larman,1999:51]



<b>Caso de uso:</b>	Nombre del caso de uso.
<b>Actores:</b>	Lista de actores, en la cual se indica quien inicia el caso de uso.
<b>Propósito:</b>	Intención del caso de uso.
<b>Resumen:</b>	Repetición del caso de uso de alto nivel o una síntesis similar.
<b>Tipo:</b>	Primario, secundario u opcional. Esencial o real.
<b>Referencias cruzadas:</b>	Casos relacionados de uso y funciones también relacionadas del sistema
<b>Curso normal de eventos:</b>	descripción detallada de la conversación interactiva entre los actores y el sistema.

*Acción del actor*

*respuesta del sistema*



## 1. Introducción

Este documento tiene como objetivo el ayudar en el entendimiento del problema al describir la funcionalidad del sistema en términos de diagramas gráficos UML (casos de uso) y de una descripción concisa en lenguaje natural de los mismos.

Específicamente se relaciona con la funcionalidad para ingresar al sistema, dado que la información no puede ser modificada por cualquier usuario, se requiere de autenticación para el administrador de la agenda.

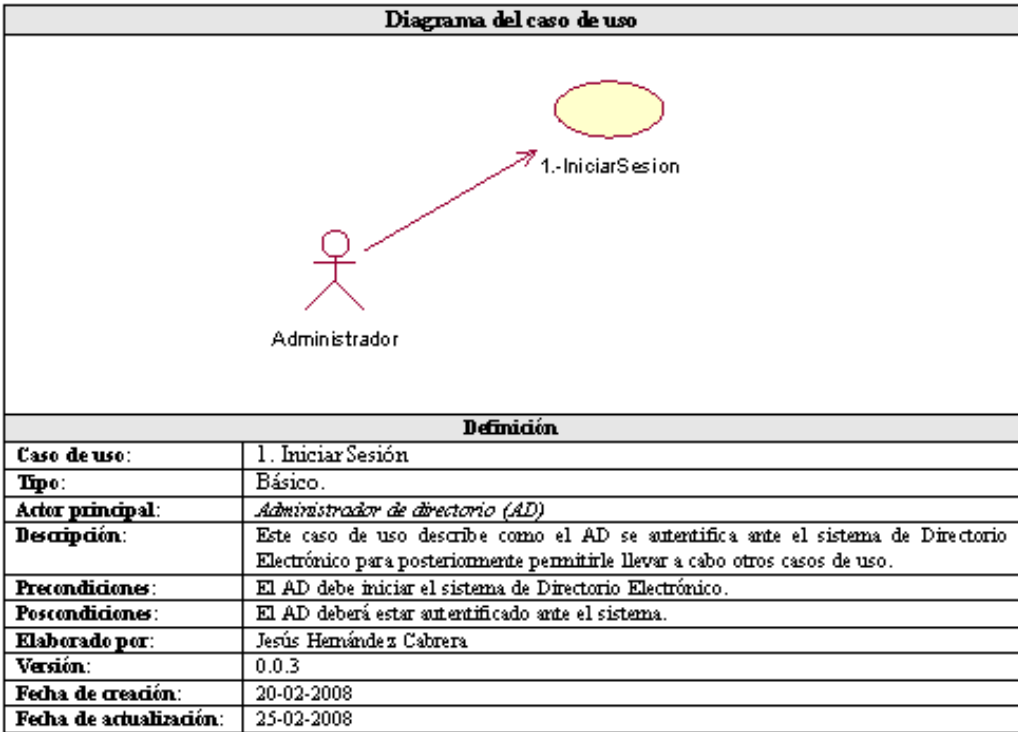
## 2. Referencias

El presente documento hace referencia a los siguientes documentos:

- Especificación de Requerimientos (2.1-EspecificacionRequerimientos.doc).

## 3. Contenido

### 3.1 Detalle del Caso de Uso “1. Registrar Cuenta”

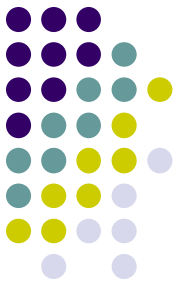




Flujos				
<b>Flujo básico:</b>				
<b>Iniciar Sesión</b>				
Inicio de sesión normal				
Actor		Sistema		
Paso	Acciones	Paso	Acciones	Excepción
1.	El <b>AD</b> elige autenticarse ante el sistema de Directorio Electrónico.	2.	El sistema muestra un cuadro de dialogo en el cual pide la contraseña del sistema.	
3	El <b>AD</b> ingresa la contraseña y pulsa el botón autenticar.	4	Verifica la información contra la contraseña almacenada en la base de datos.	E1
		5	Si la contraseña es correcta el sistema habilitará las opciones: <ul style="list-style-type: none"> <li>• Registrar contacto</li> <li>• Eliminar contacto</li> <li>• Modificar contacto</li> </ul>	

<b>Flujos alternativos:</b>				
<b>Iniciar Sesión, contraseña inválida</b>				
Actor		Sistema		
Paso	Acciones	Paso	Acciones	Excepción
1.	El <b>AD</b> elige autenticarse ante el sistema de Directorio Electrónico.	2.	El sistema muestra un cuadro de dialogo en el cual pide la contraseña del sistema.	
3	El <b>AD</b> ingresa la contraseña y pulsa el botón autenticar.	4	Verifica la información contra la contraseña almacenada en la base de datos.	
		5	Si la contraseña proporcionada no es la correcta o esta vacía mostrar un cuadro de dialogo de error.	

Excepciones		
Identificador	Nombre	Respuesta del Sistema
E1	Conexión con BD perdida	Cuadro de dialogo que muestre la siguiente leyenda: "Verifique que hay una conexión a Base de Datos"



# Bibliografía y créditos

- Larman Craig, UML y patrones, introducción al análisis y diseño orientado a objetos; México, Prentice Hall, 1999.
- Ana Maria Briseño. Ingeniería de requerimientos, material de la Maestría en ciencias de la computación de la UNAM.